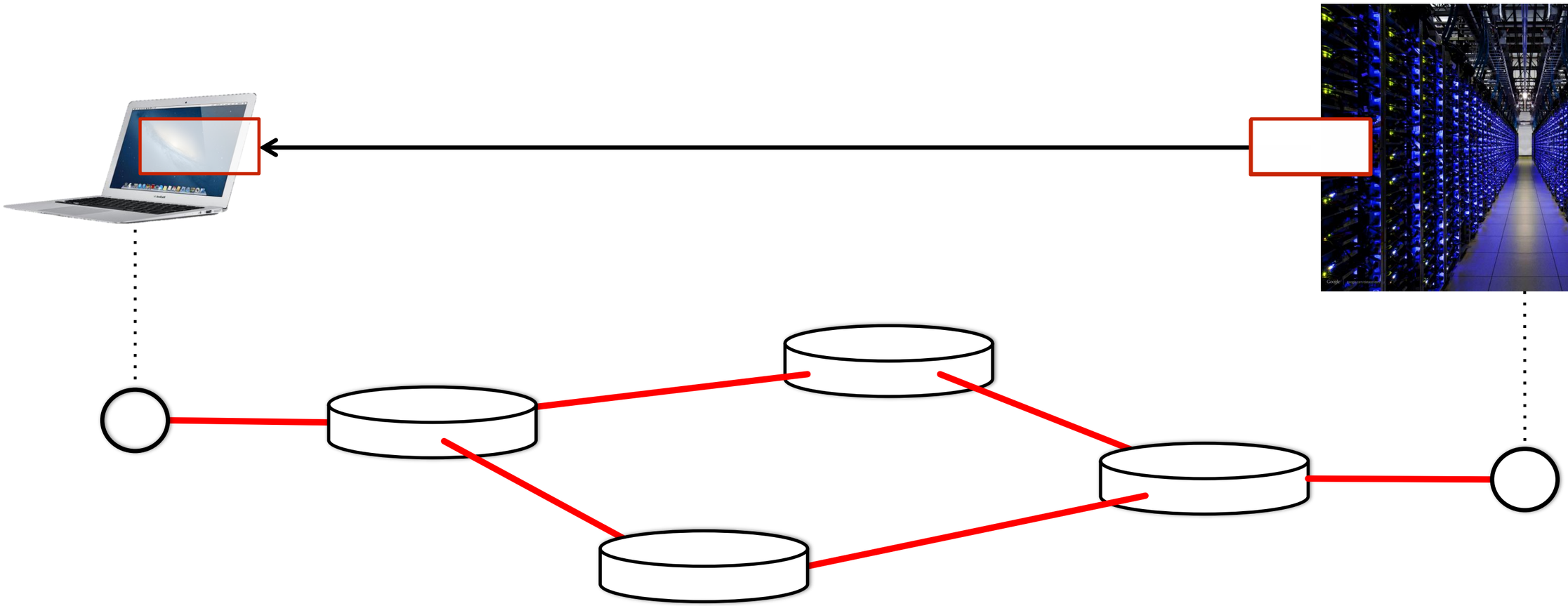
How the Internet Works

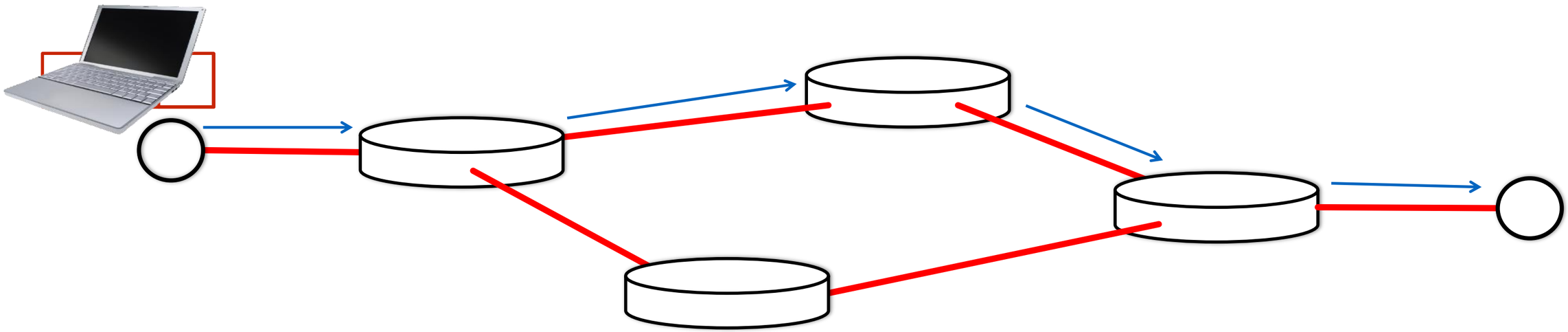
(in 50 mins)

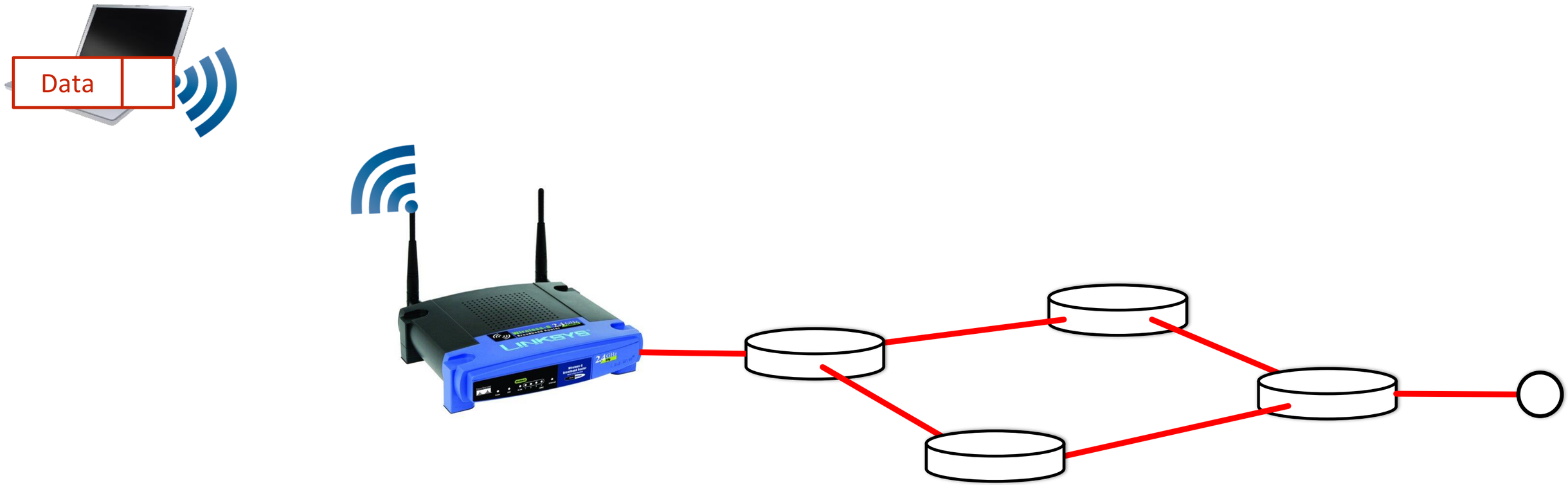
Nick McKeown
Professor of Electrical Engineering
and Computer Science, Stanford University

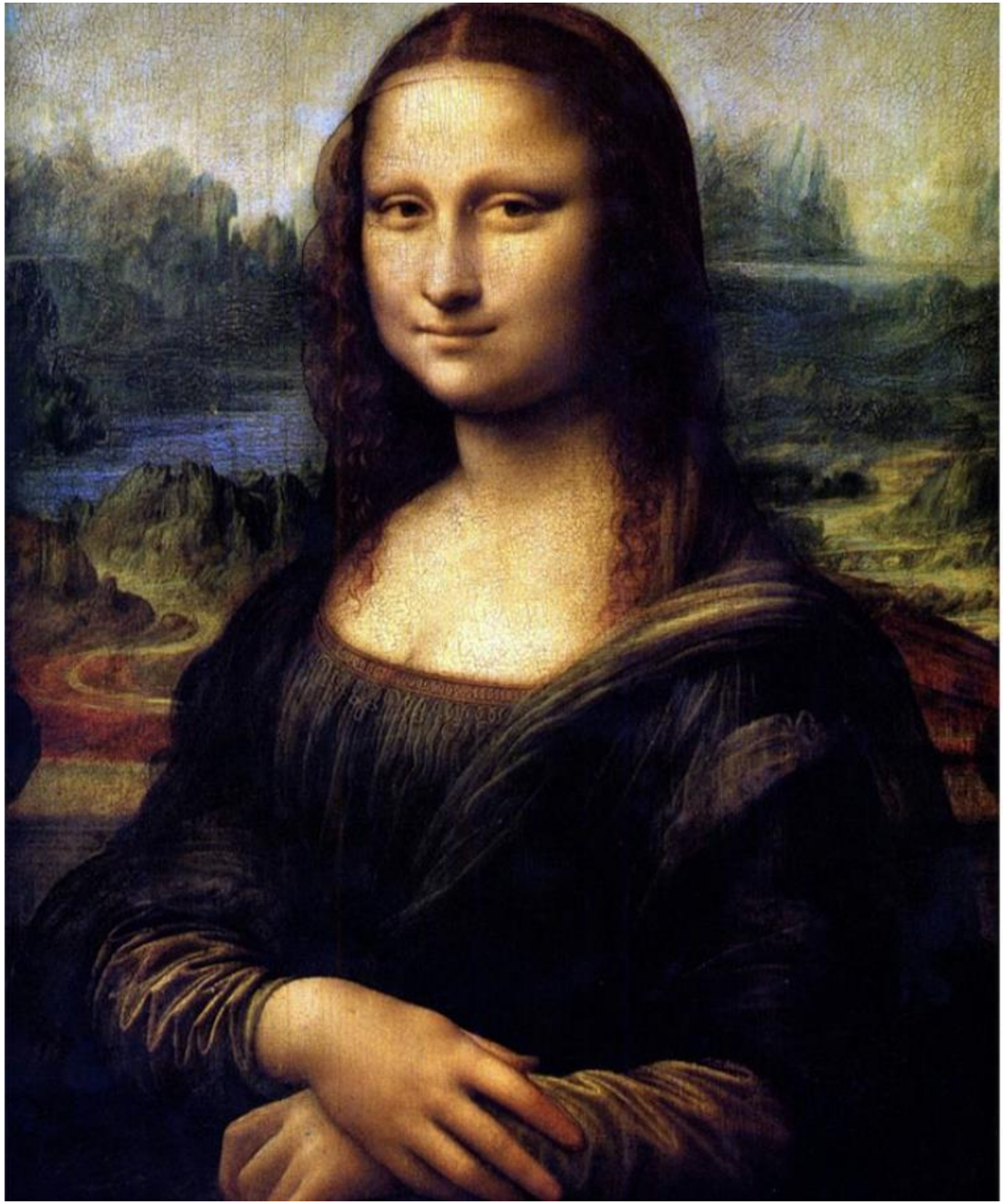
Google NETFLIX www.stanford.edu

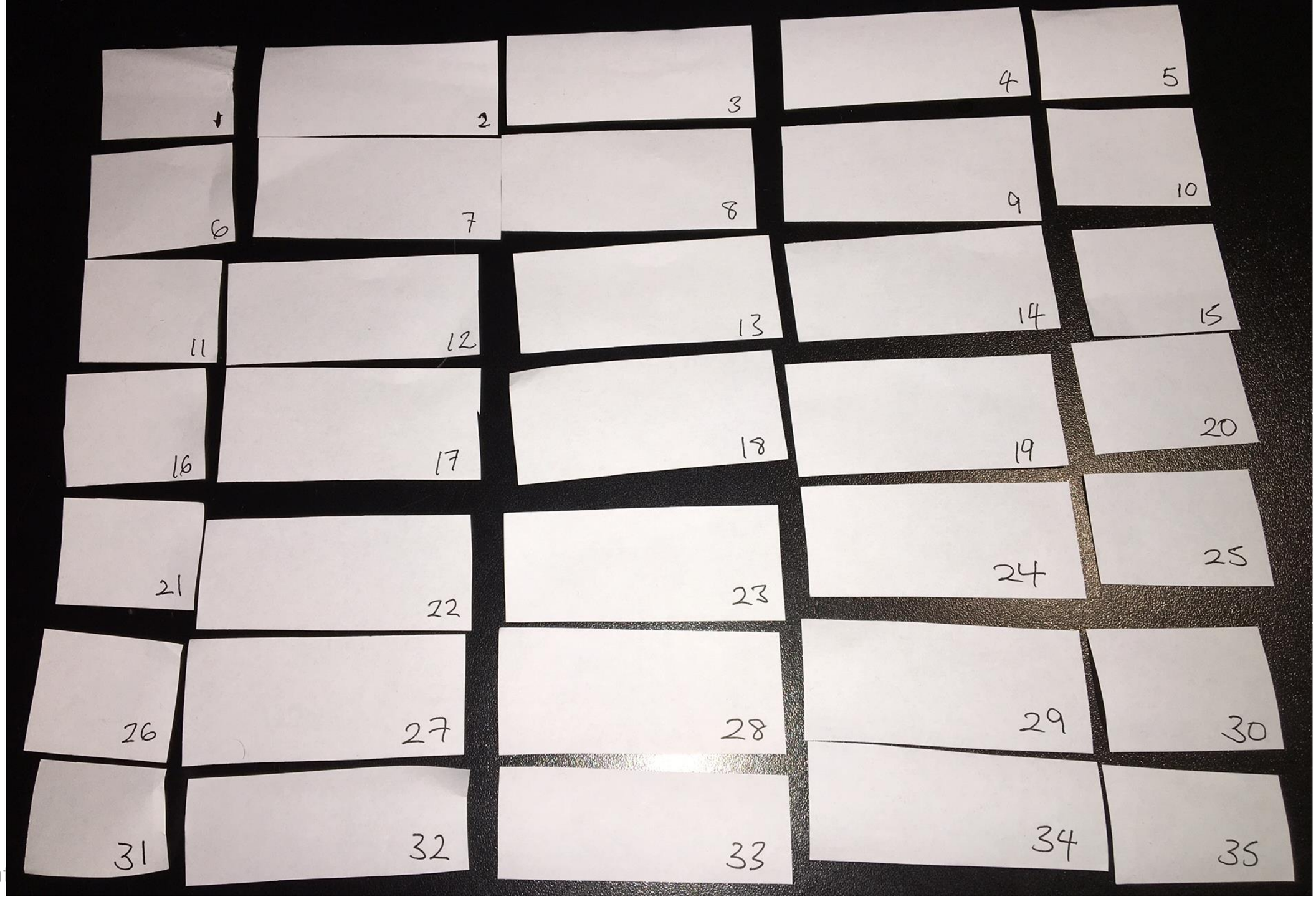



My Program
Someone else's Program

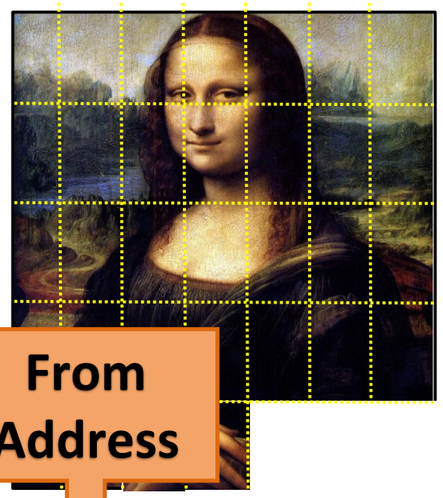



Data
LINKSYS

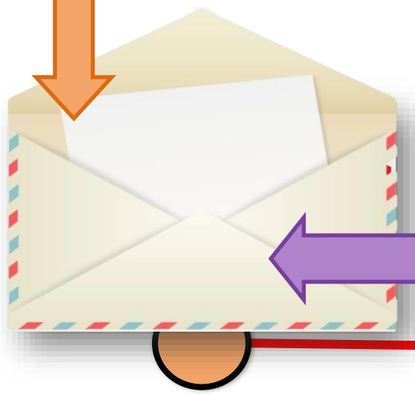



1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35

A more accurate model



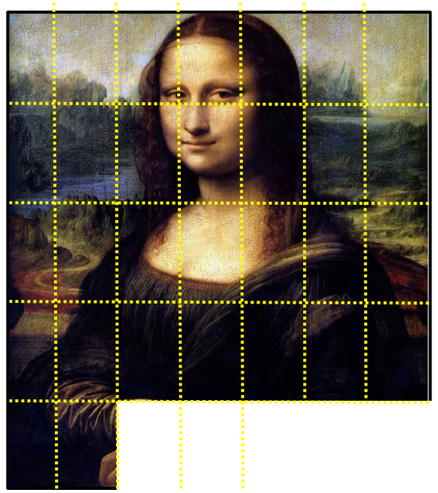
From Address



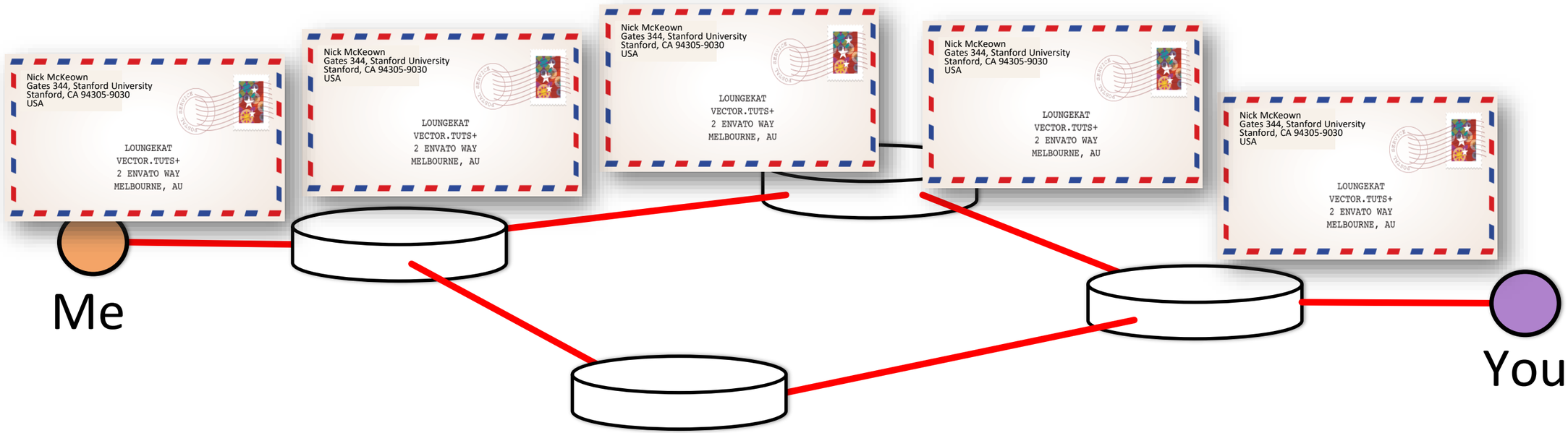
To Address

Me

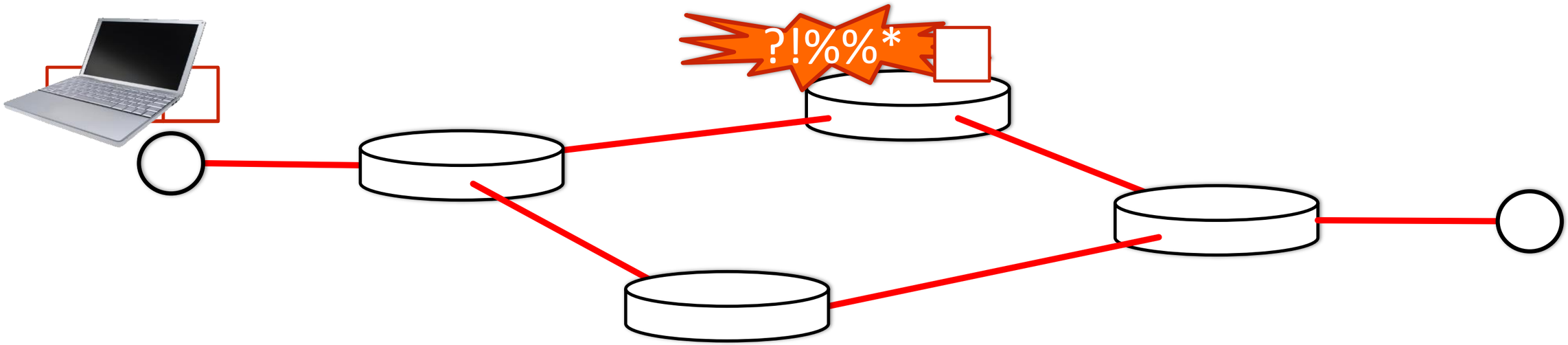
You



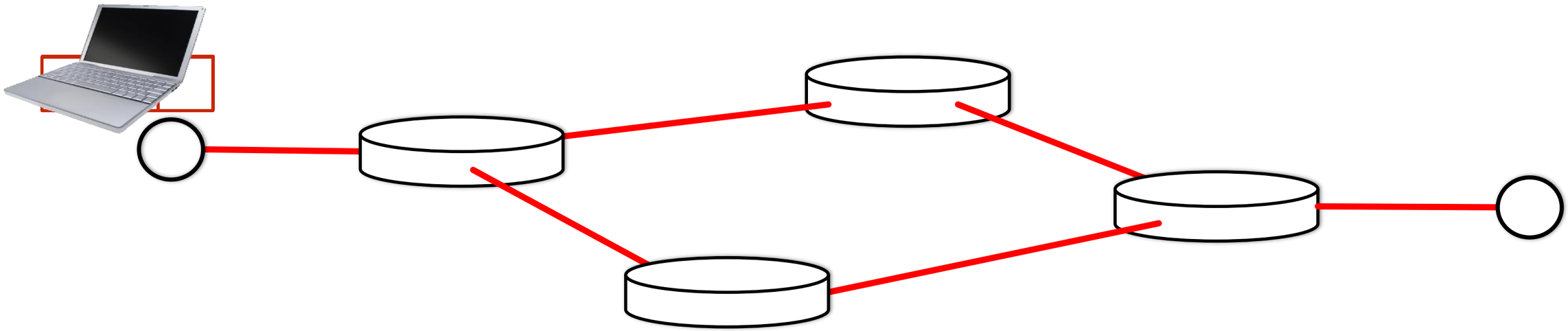
Multiple packets in pipeline at same time



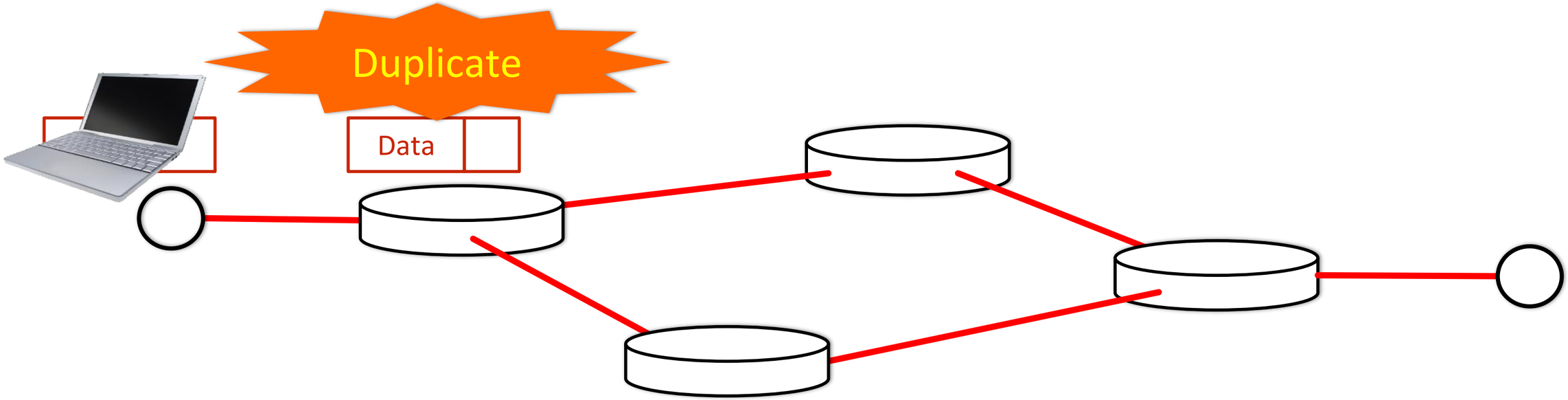
Packets may be damaged



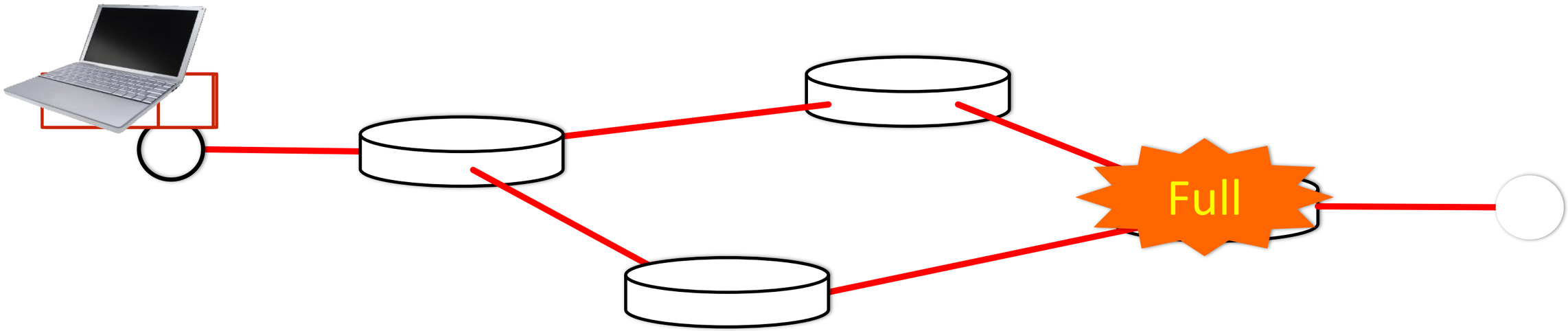
Packets may arrive out of order



Packets may be duplicated



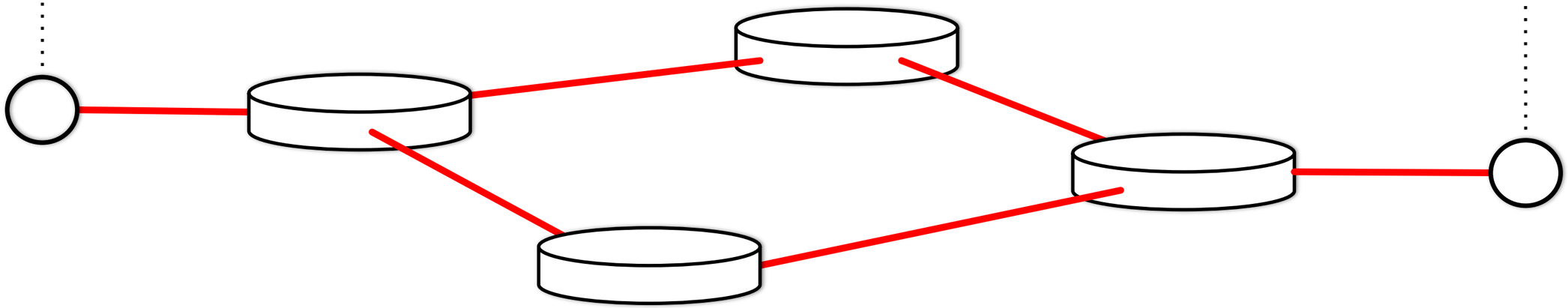
They may not arrive at all!



Summary so far

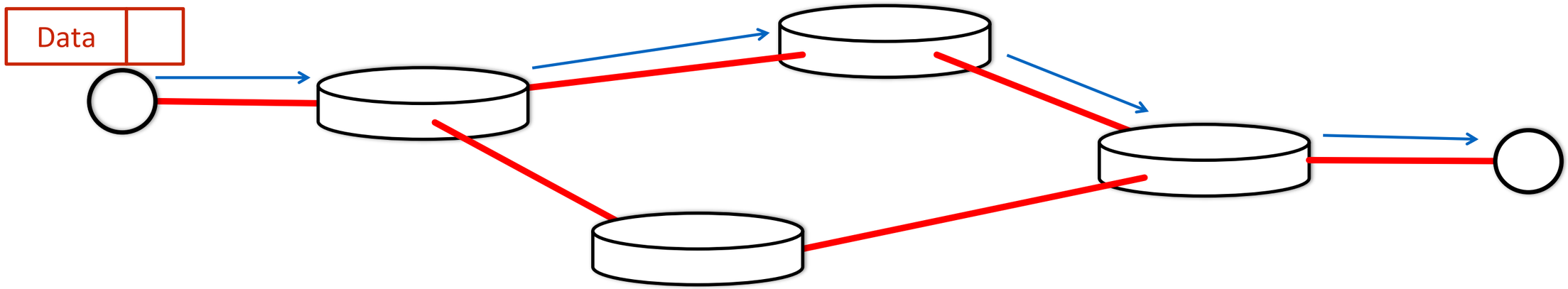
Applications send and receive data in packets….

…over an Internet that is unreliable.



How packets find their way across the Internet

Internet addresses



Internet Addresses (“IP address”)

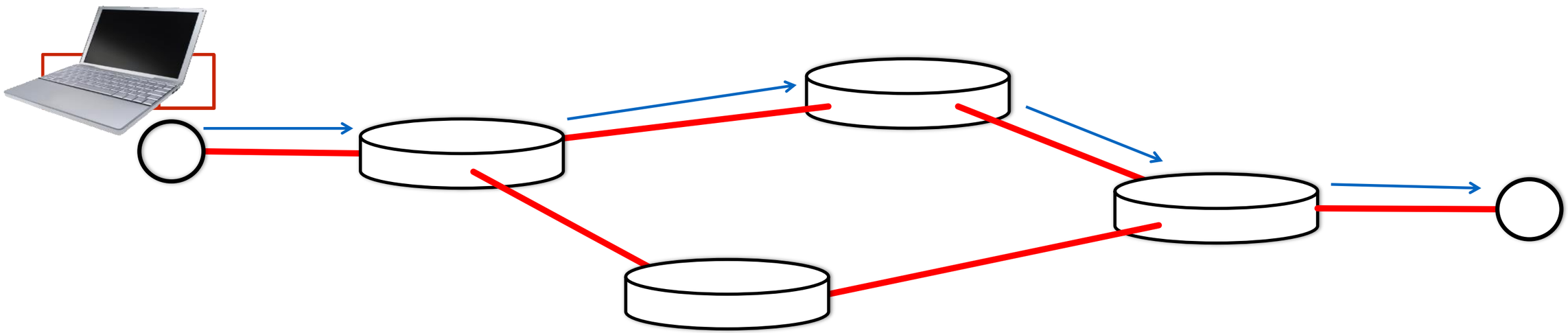
All Internet packets carry a destination IP address.

We usually write the IP address like this:

171.64.74.58

Routers forward packets **one at a time**.

Routers look at IP addresses, then send packets to a router closer to the destination.



IP Addresses

The IP address tells a router where to send the packet next.

IP addresses have *structure*

A network interface on a computer at Stanford University

171.64.74.58

An address managed by Stanford (Stanford University Network)

88.255.96.208

Can we see the path our packets take?

Yes!

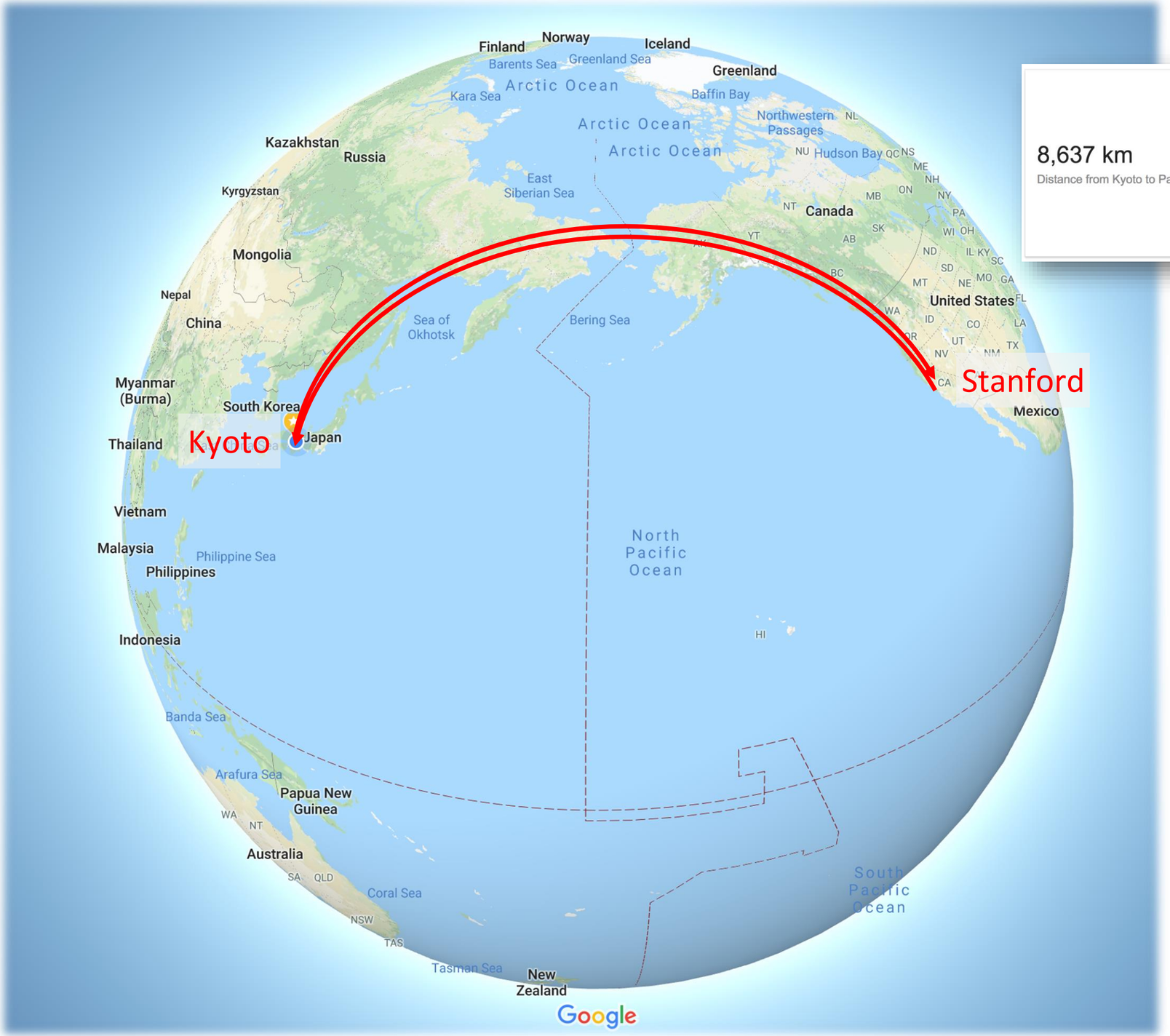
On your computer, try: “`ping yuba.stanford.edu`”

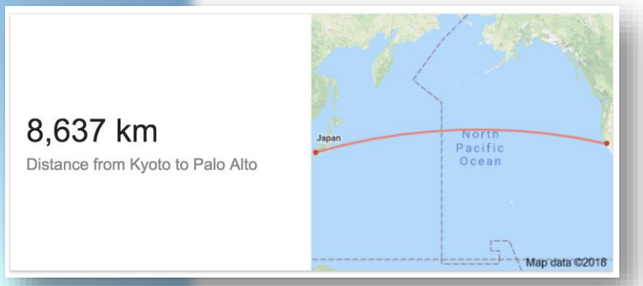
and “`traceroute yuba.stanford.edu`”

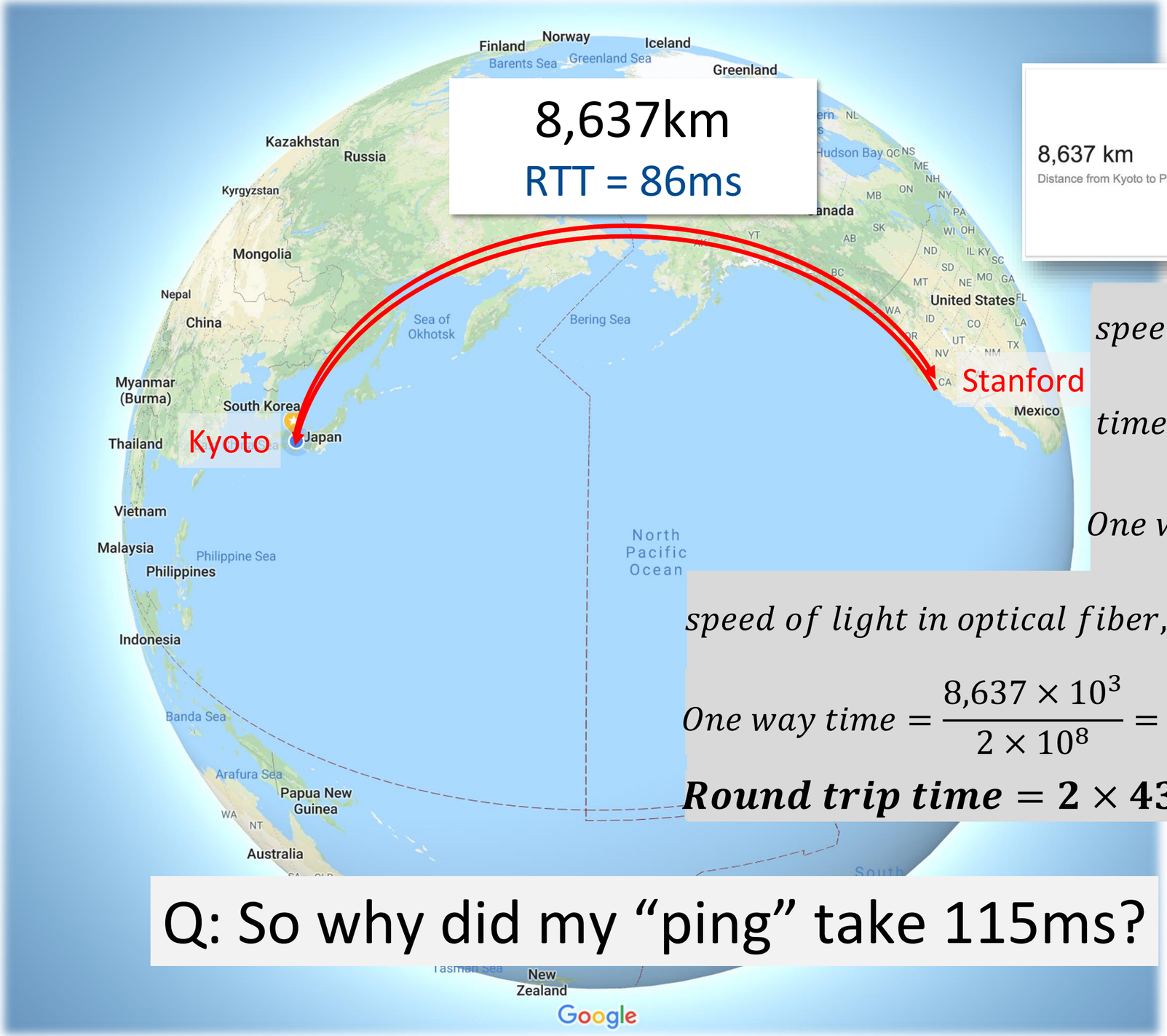
(Windows: “`tracert yuba.stanford.edu`”)

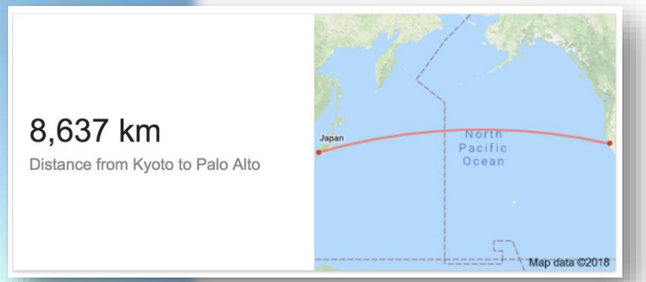
```
ping www.doshisha.ac.jp
```

From yuba (at Stanford), it takes about 115ms to reach Japan and back again
(“round-trip-time”)


Finland
Norway
Iceland
Barents Sea
Greenland Sea
Greenland
Arctic Ocean
Kara Sea
Baffin Bay
Arctic Ocean
Northwestern Passages
Arctic Ocean
Kazakhstan
Russia
Hudson Bay
East Siberian Sea
Kyrgyzstan
Canada
Mongolia
Nepal
China
Sea of Okhotsk
Bering Sea
United States
Myanmar (Burma)
South Korea
Stanford
Mexico
Kyoto
Japan
Thailand
Vietnam
Malaysia
Philippine Sea
Philippines
North Pacific Ocean
Indonesia
Banda Sea
Arafura Sea
Papua New Guinea
Australia
Coral Sea
South Pacific Ocean
Tasman Sea
New Zealand
Google


8,637 km
Distance from Kyoto to Palo Alto
Japan
North Pacific Ocean
Map data ©2018





$$speed = \frac{distance}{time}$$

$$time = \frac{distance}{speed}$$

$$One\ way\ time = \frac{8{,}637 \times 10^3}{c'}$$

$$speed\ of\ light\ in\ optical\ fiber, c' \approx \frac{2}{3}c = 2 \times 10^8 m/s$$

$$One\ way\ time = \frac{8{,}637 \times 10^3}{2 \times 10^8} = 43ms$$

$$\boldsymbol{Round\ trip\ time = 2 \times 43ms = 86ms}$$

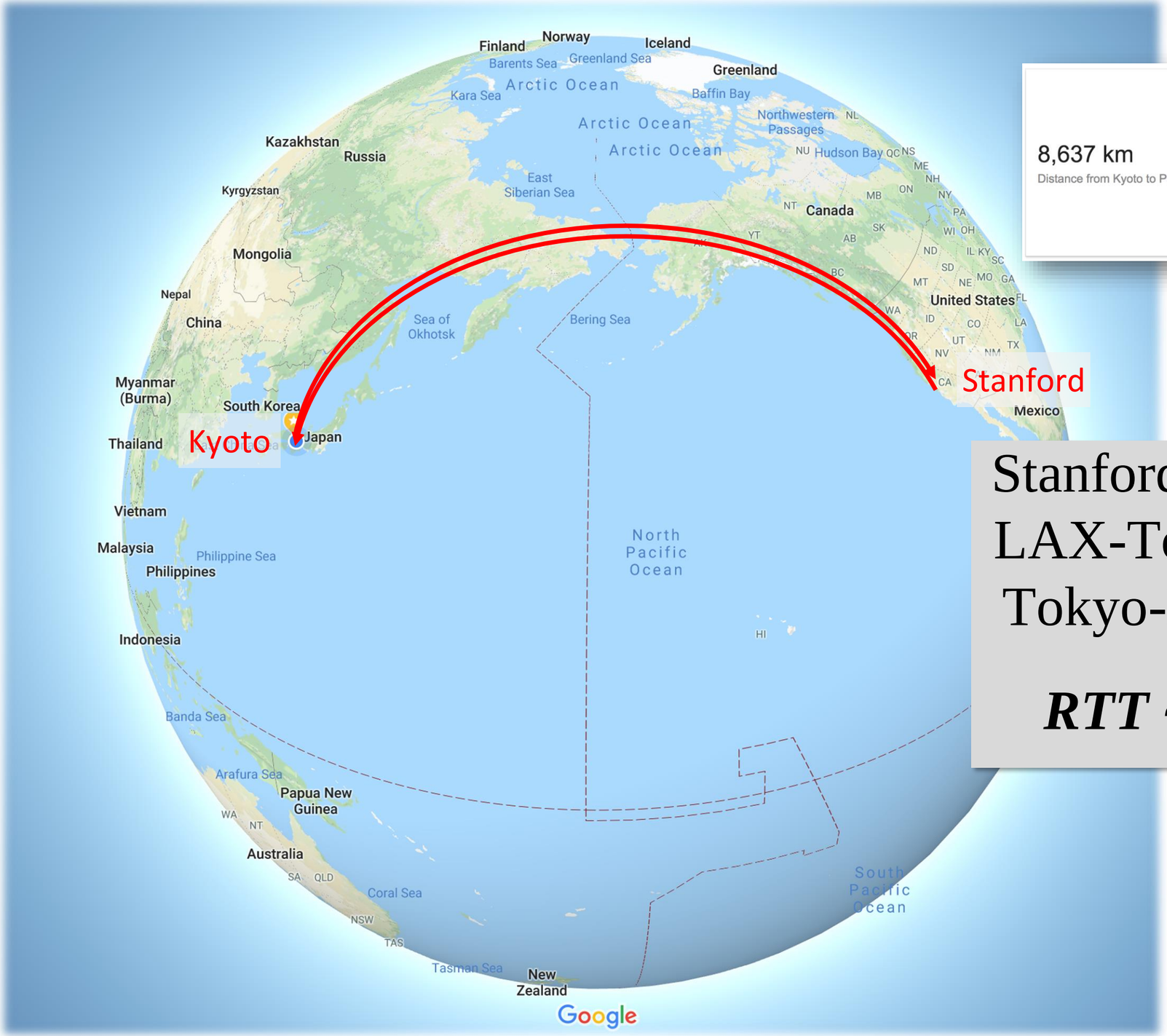
Q: So why did my “ping” take 115ms?

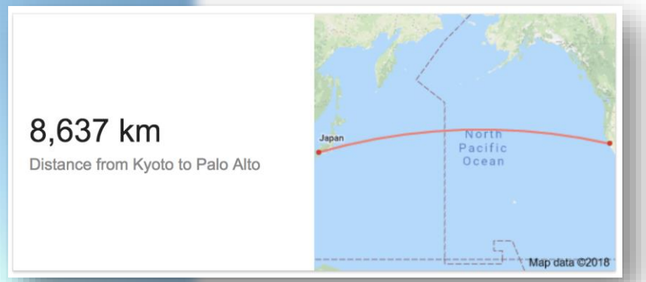
traceroute

From Stanford to Doshisha University (Kyoto)

nickm@yuba.Stanford.EDU > **traceroute -q1 istc.doshisha.ac.jp**

```
traceroute to istc.doshisha.ac.jp (202.23.190.159), 30 hops max, 40 byte
packets
 1  csmx-west-rtr.SUNet (171.64.74.2)  0.234 ms
 2  hpr-svl-rtr-vlan3.SUNet (171.66.255.186)  0.423 ms
 3  hpr-svl-hpr2--stan-ge.cenic.net (137.164.27.161)  0.779 ms
 4  hpr-lax-hpr3--svl-hpr3-100ge.cenic.net (137.164.25.73)  8.627 ms
 5  sinet-1-lo-jmb-702.lsanca.pacificwave.net (207.231.240.135)  8.793 ms
 6  tokyo1-GM-ET-8-3-0-100.s5.sinet.ad.jp (150.99.89.242)  108.852 ms
 7  kyoto-RM-ET-7-1-0-1151.s5.sinet.ad.jp (150.99.89.184)  114.829 ms
 8  doshisha-LAN.gw.sinet.ad.jp (150.99.196.66)  115.249 ms
 9  ***
```



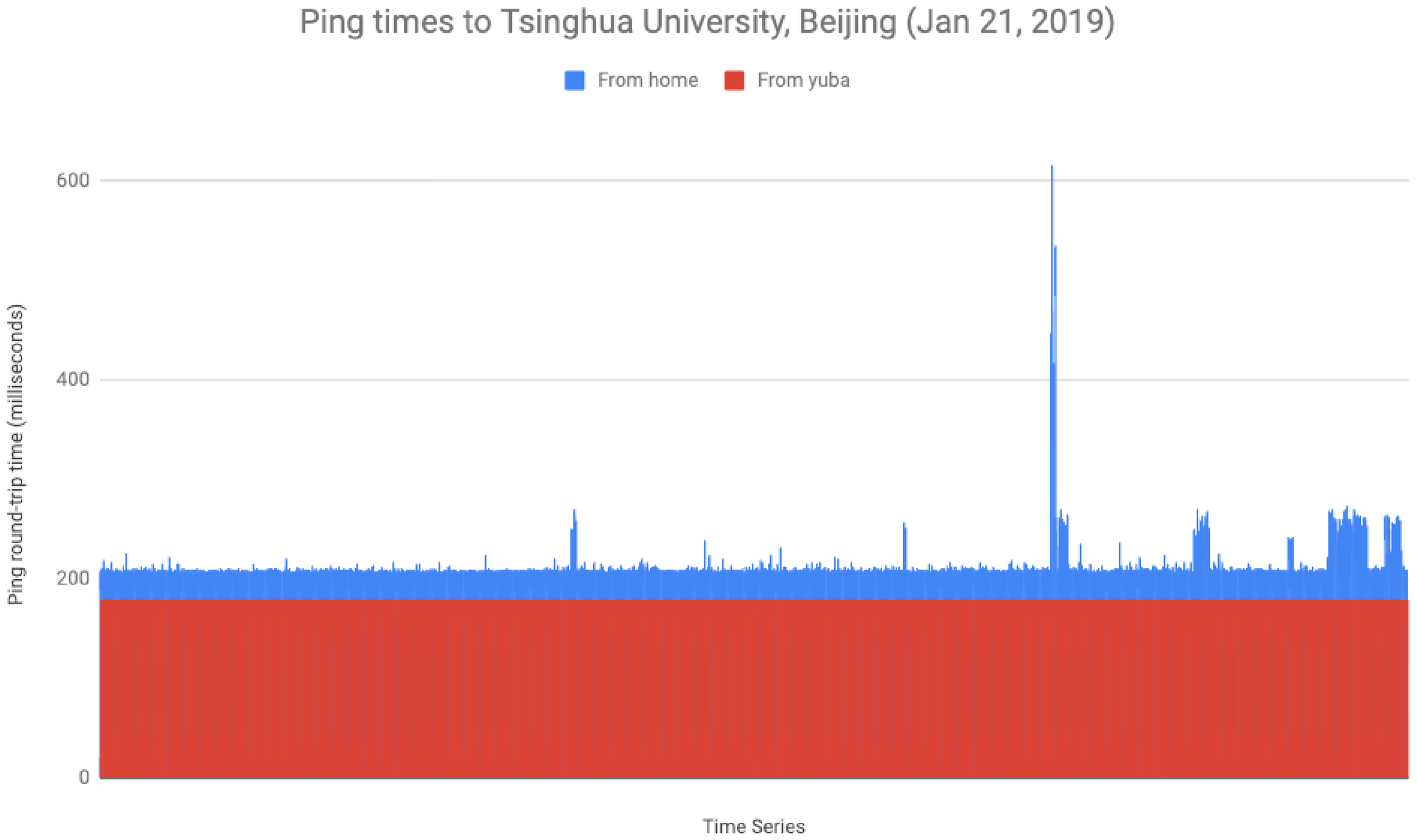


Stanford-LAX: 500km

LAX-Tokyo: 8,800km

Tokyo-Kyoto: 500km

RTT ~= 100ms


Ping times to Tsinghua University, Beijing (Jan 21, 2019)
From home
From yuba
600
400
200
0
Ping round-trip time (milliseconds)
Time Series

Try `traceroute` to….

yuba.stanford.edu, www.google.com, www.ntua.gr, …

```
nickm @ home on Comcast network $ traceroute -q1 www.ntua.gr
traceroute to www.ntua.gr (147.102.224.101), 64 hops max, 52 byte packets
 1  testwifi.here (192.168.86.1)  0.921 ms
 2  96.120.91.229 (96.120.91.229)  8.491 ms
 3  be-20052-rur02.santaclara.ca.sfba.comcast.net (68.87.196.49)  10.237 ms
 4  162.151.78.129 (162.151.78.129)  8.857 ms
 5  be-232-rar01.santaclara.ca.sfba.comcast.net (162.151.78.253)  8.941 ms
 6  be-3651-cr02.sunnyvale.ca.ibone.comcast.net (68.86.91.73)  9.485 ms
 7  be-11083-pe02.529bryant.ca.ibone.comcast.net (68.86.84.14)  10.357 ms
 8  50.248.118.238 (50.248.118.238)  9.467 ms
 9  be2016.ccr22.sfo01.atlas.cogentco.com (154.54.0.177)  11.134 ms
10  be3110.ccr32.slc01.atlas.cogentco.com (154.54.44.142)  35.123 ms
11  be3038.ccr22.den01.atlas.cogentco.com (154.54.42.98)  35.458 ms
12  be3036.ccr22.mci01.atlas.cogentco.com (154.54.31.90)  53.296 ms
13  be2832.ccr42.ord01.atlas.cogentco.com (154.54.44.170)  58.615 ms
14  be2718.ccr22.cle04.atlas.cogentco.com (154.54.7.130)  65.101 ms
15  be2994.ccr32.yyz02.atlas.cogentco.com (154.54.31.234)  73.458 ms
16  be3260.ccr22.ymq01.atlas.cogentco.com (154.54.42.90)  85.407 ms
17  be3043.ccr22.lpl01.atlas.cogentco.com (154.54.44.165)  150.394 ms
18  be2183.ccr42.ams03.atlas.cogentco.com (154.54.58.70)  156.881 ms
19  be2814.ccr42.fra03.atlas.cogentco.com (130.117.0.142)  162.831 ms
20  be2960.ccr22.muc03.atlas.cogentco.com (154.54.36.254)  169.709 ms
21  be3462.ccr52.vie01.atlas.cogentco.com (154.54.59.181)  173.362 ms
22  149.6.175.250 (149.6.175.250)  173.362 ms
23  ae1.mx1.ath2.gr.geant.net (62.40.98.146)  200.320 ms
24  grnet-ias-grnet-gw.mx1.ath2.gr.geant.net (83.97.88.66)  206.457 ms
25  eier-kolettir-ae.backbone.grnet.gr (62.217.100.63)  208.703 ms
26  ntua-zogr-3.eier.access-link.grnet.gr (62.217.96.169)  201.891 ms
27  *
```

```
nickm@yuba.Stanford.EDU > traceroute -q1 www.ntua.gr
traceroute to www.ntua.gr (147.102.224.101), 30 hops max, 40 byte packets
 1  csee-west-rtr-vl3874.SUNet (171.64.74.2)  0.289 ms
 2  hpr-svl-rtr-vlan2.SUNet (171.64.255.147)  0.637 ms
 3  hpr-svl-hpr3--stan-100ge.cenic.net (137.164.27.60)  0.657 ms
 4  hpr-lax-hpr3--svl-hpr3-100ge.cenic.net (137.164.25.73)  8.558 ms
 5  hpr-i2--lax-hpr3-r-and-e.cenic.net (137.164.26.201)  8.723 ms
 6  ae-5.4079.rtsw.wash.net.internet2.edu (162.252.70.158)  68.694 ms
 7  internet2-gw.mx1.lon.uk.geant.net (62.40.124.44)  143.377 ms
 8  ae6.mx1.lon2.uk.geant.net (62.40.98.37)  144.475 ms
 9  ae5.mx1.par.fr.geant.net (62.40.98.179)  151.959 ms
10  ae5.mx1.gen.ch.geant.net (62.40.98.182)  158.004 ms
11  ae4.mx1.mil2.it.geant.net (62.40.98.89)  164.868 ms
12  ae3.mx2.ath.gr.geant.net (62.40.98.151)  187.210 ms
13  grnet-gw.mx2.ath.gr.geant.net (62.40.124.90)  186.227 ms
14  ntua-zogr-3.eier.access-link.grnet.gr (62.217.96.169)  188.246 ms
15  *
```

Competition (optional)

Task: Find the longest loop-free traceroute in the Internet

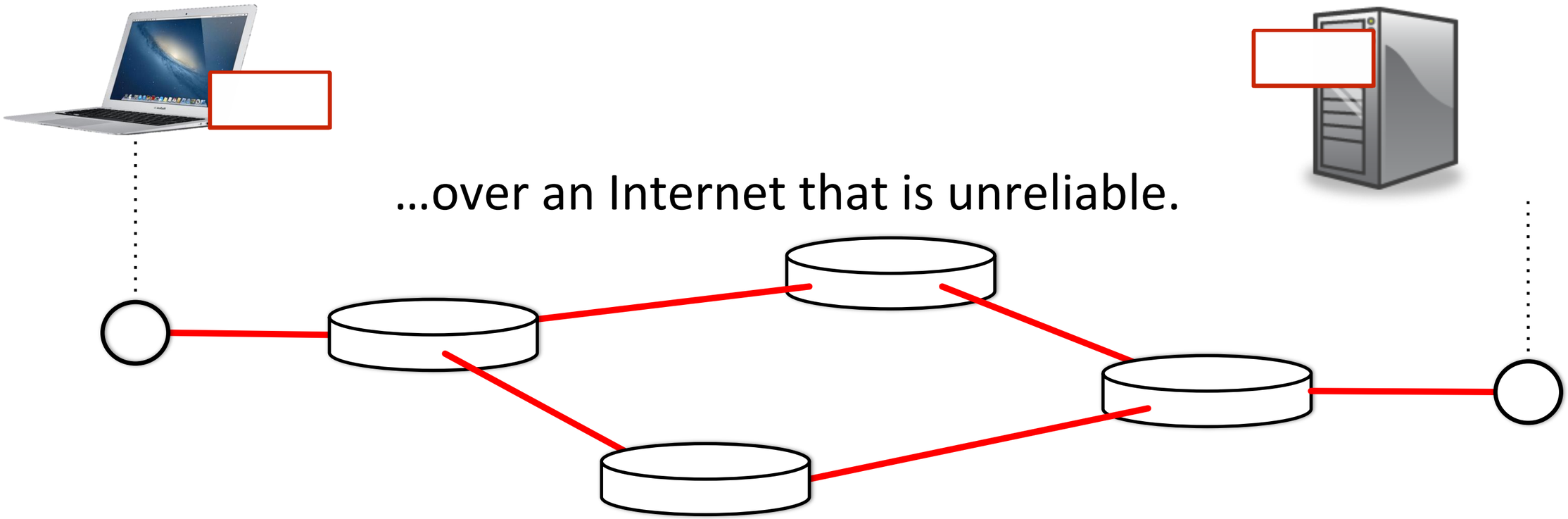
Rules

1. We must see it! So it must either be reproducible by the teaching staff, or you can send us a video screencast.
2. The source and destination can be any addresses you can reach.
3. At most one hop may return a "*" (i.e. not respond)
4. Packets must flow over physical links and through real routers. i.e. not a virtual, simulated or emulated network.
5. You must turn in your solution by Friday Sept 18th at 7pm Stanford time.

Prize: A small prize and big bragging rights!

Summary so far

Applications send and receive data in packets….



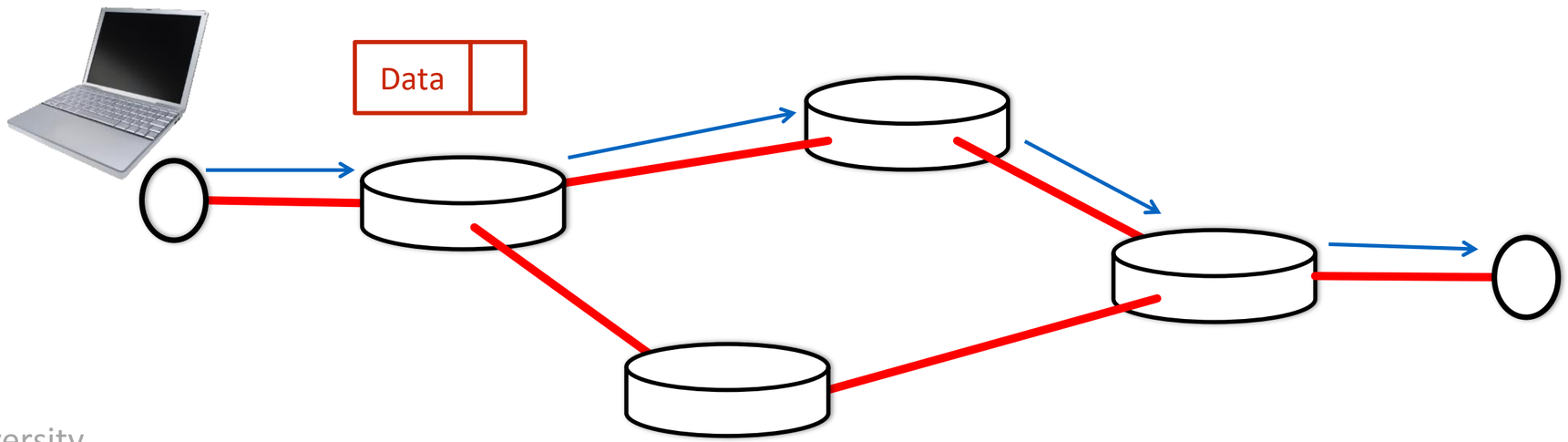
…over an Internet that is unreliable.

Packets are forwarded hop-by-hop based on the final destination address.

The Internet cannot be trusted!!

The Internet doesn't promise to deliver packets in order.
It doesn't promise to deliver packets quickly, or on time.
It doesn't even promise to deliver them at all!

It just makes a "best-effort" attempt.



Sending data reliably over an Internet that is unreliable

How Network Applications Communicate

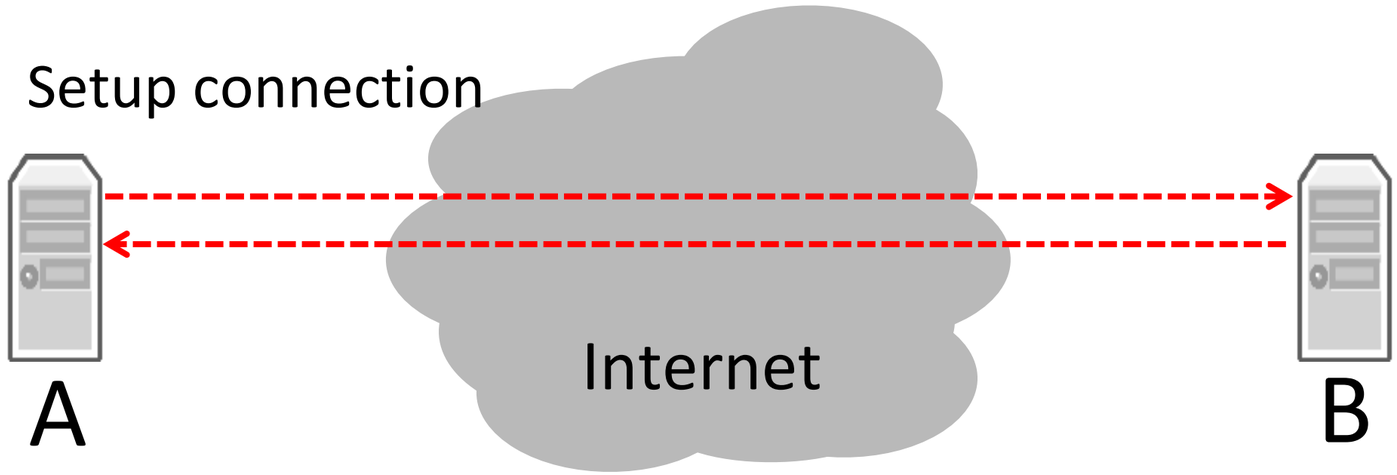
The most common method:

- Communication is in both directions – "bidirectional".
- Communication is reliable
  (if there is a working path between the two computers).

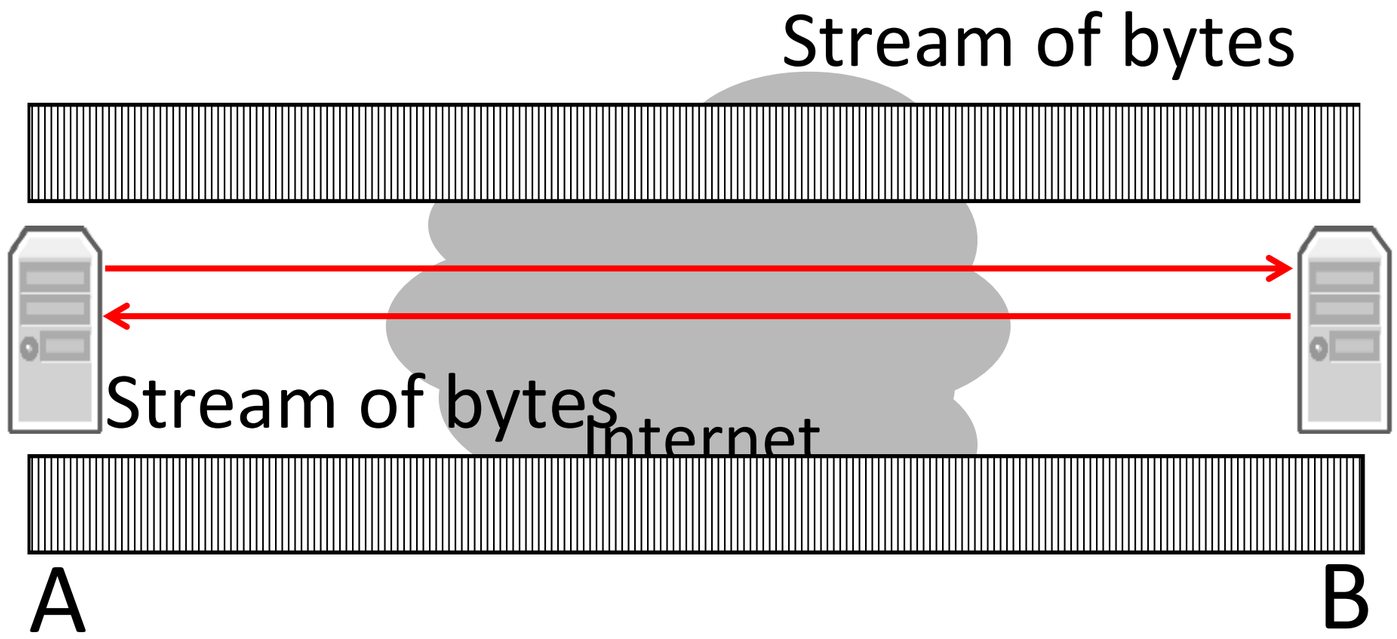
It's like an unformatted pipe:

- You push data in at one end, and it pops out correctly at the other end.
- The applications decide how the data is formatted inside the pipe.

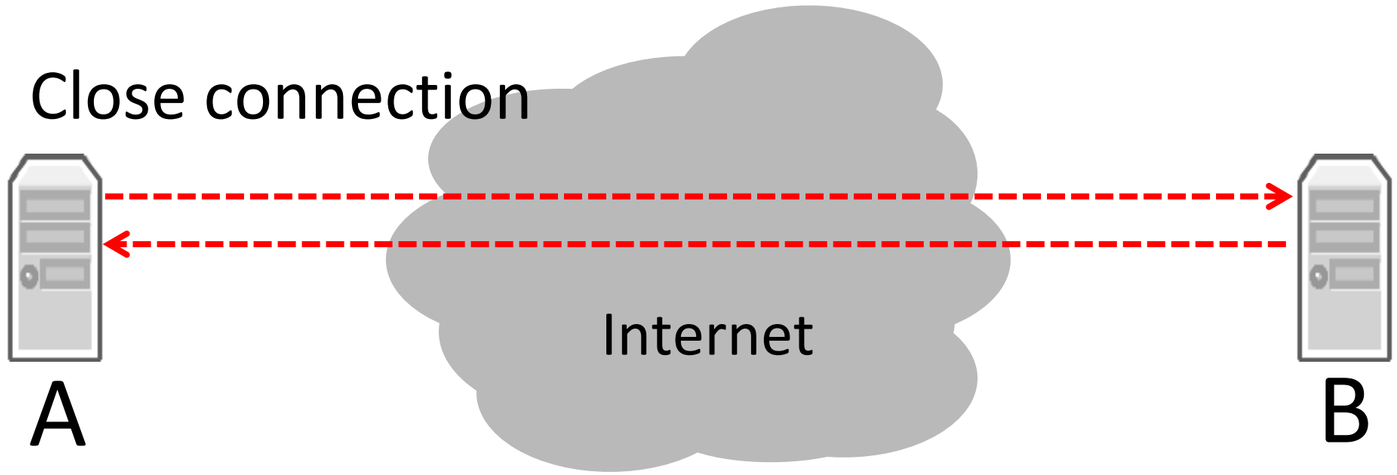
Byte Stream Model



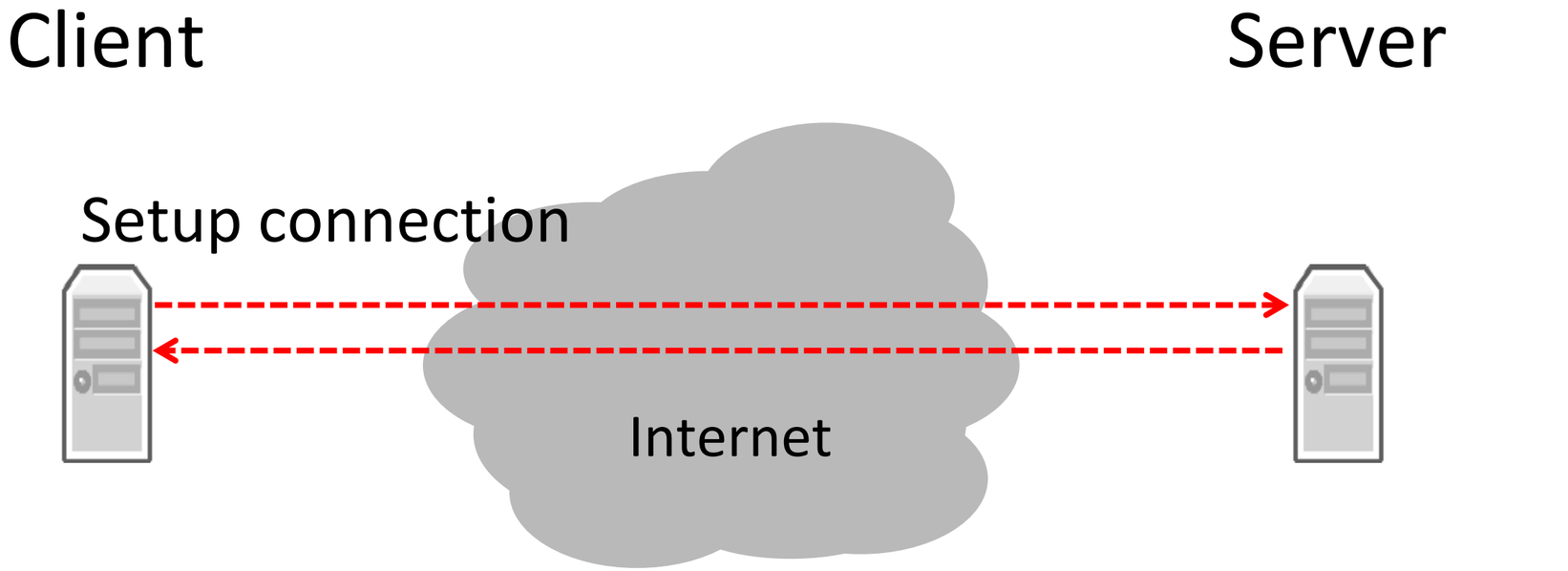
Byte Stream Model



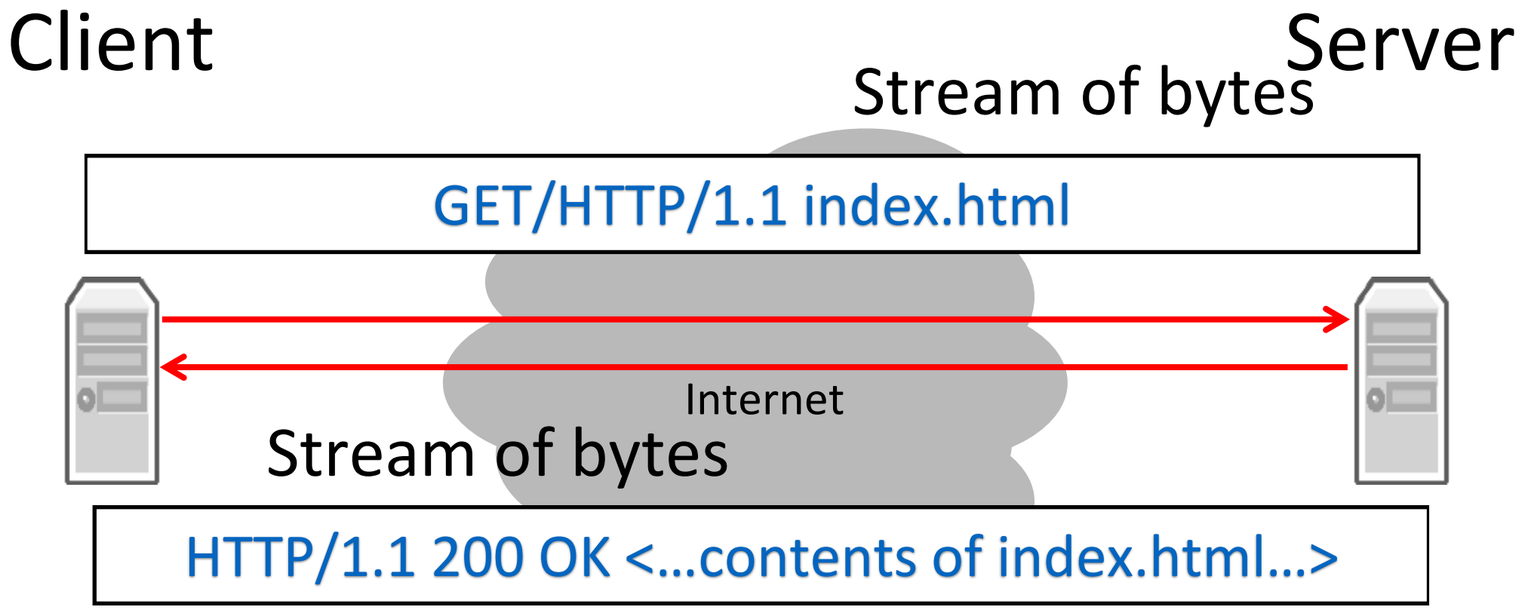
Byte Stream Model



World Wide Web (HTTP)

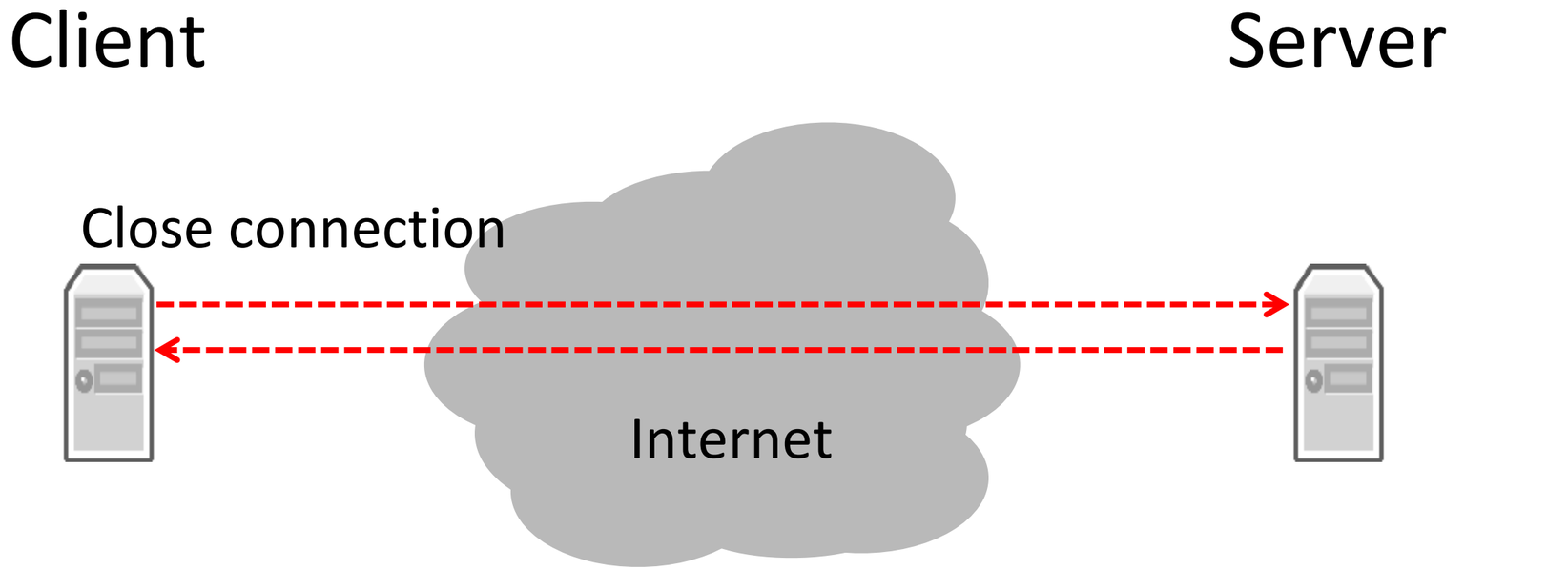


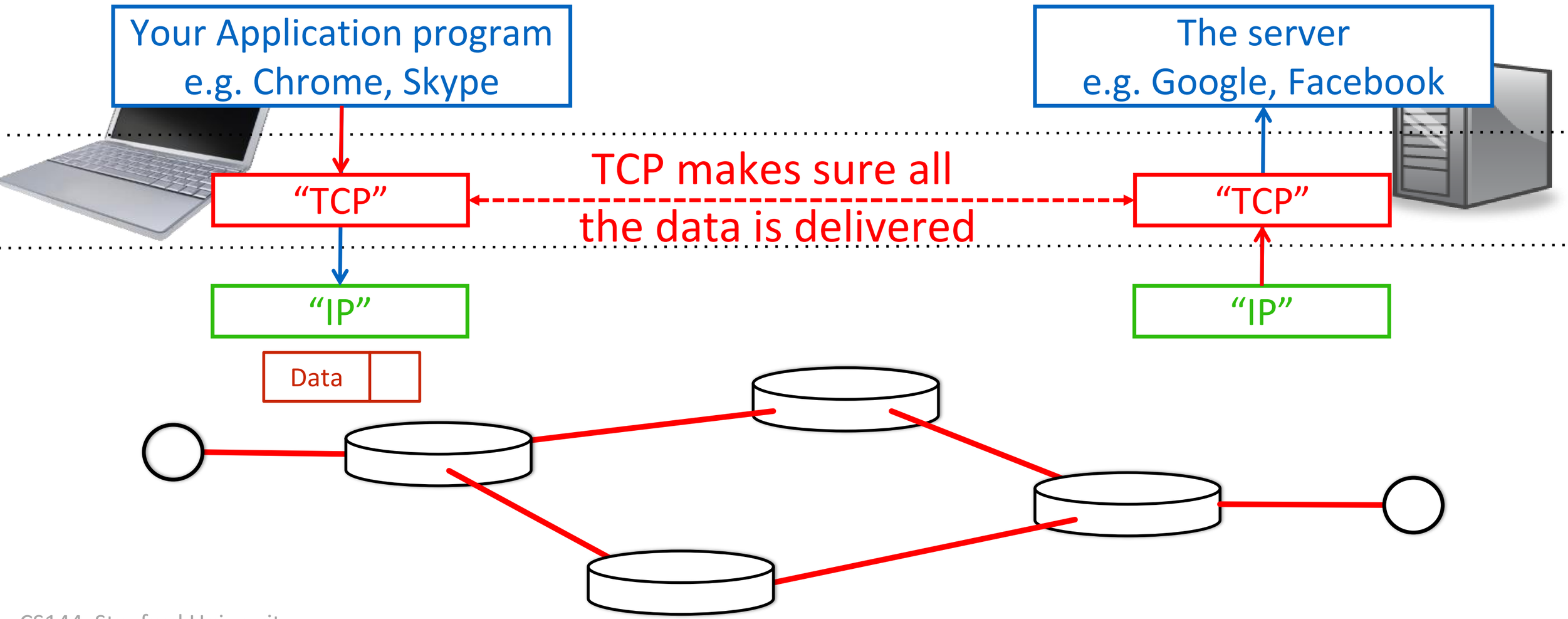
World Wide Web (HTTP)



www.stanford.edu

World Wide Web (HTTP)




Your Application program
e.g. Chrome, Skype
The server
e.g. Google, Facebook
"TCP"
TCP makes sure all
the data is delivered
"TCP"
"IP"
"IP"
Data

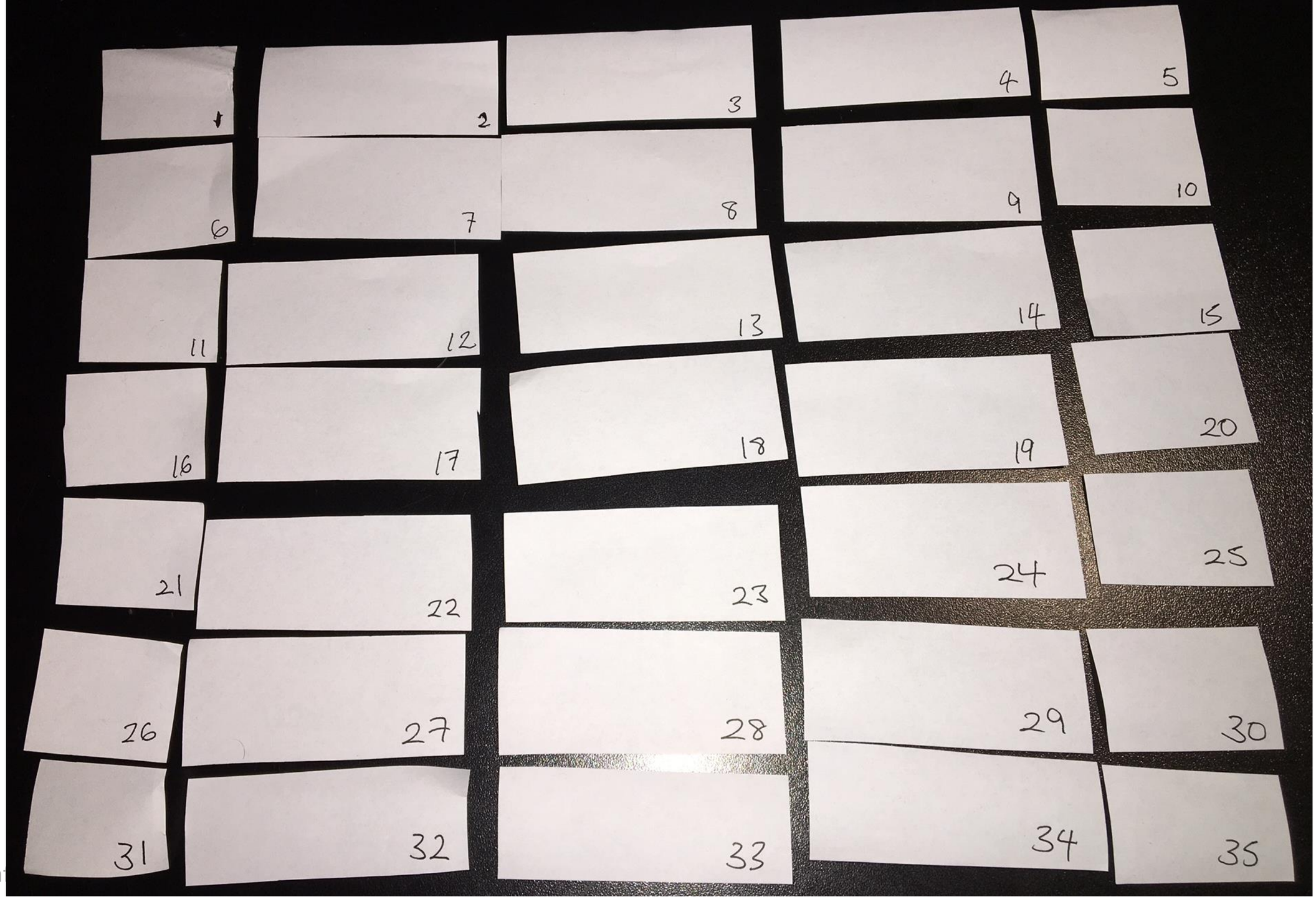
TCP's job

Makes sure all data is delivered correctly.

Delivers data to the application in the right order.

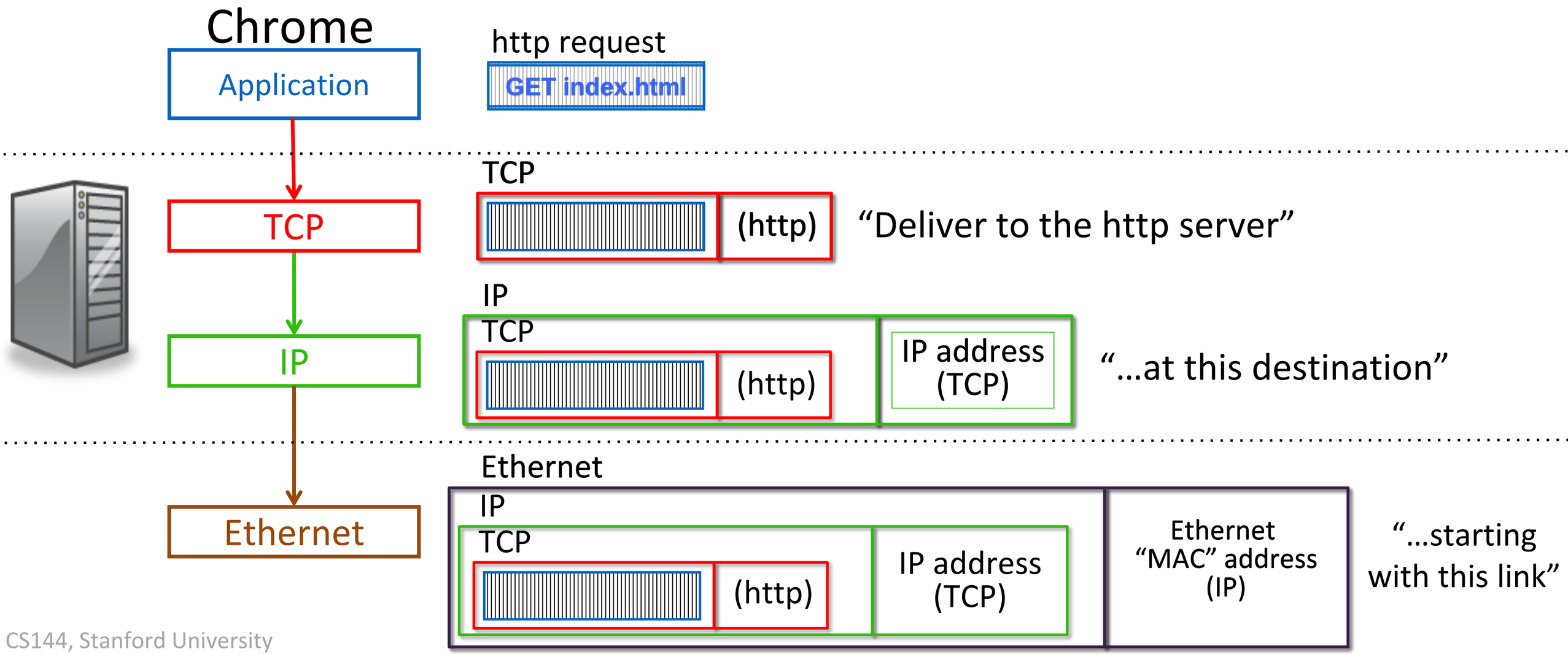
How?

- Add sequence numbers to every packet (so the receiver can check if any are missing, and put them in right order)
- When a packet arrives, send an **acknowledgment of receipt** or "ACK" back to the sender
- If no acknowledgment is received, resend the data

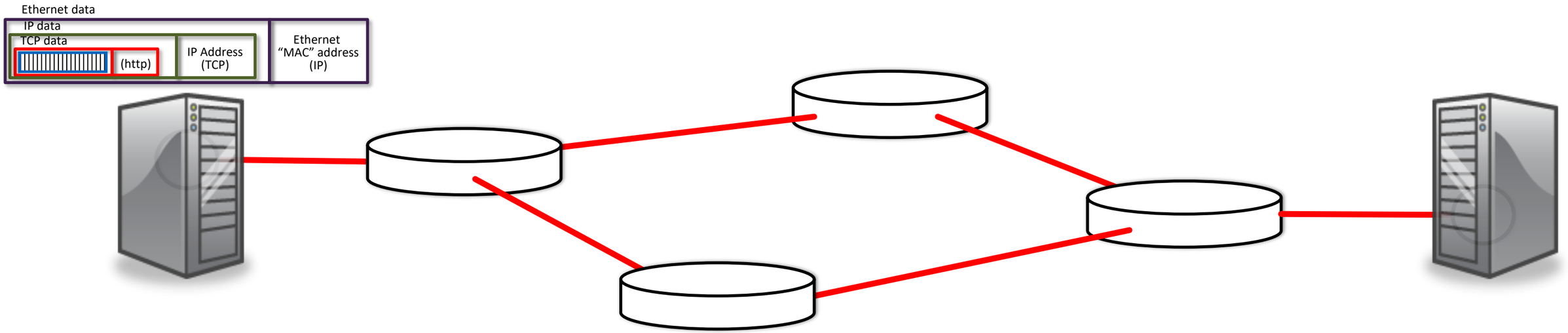

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35

Packets are “encapsulated” by different “layers” of processing

http client (e.g. Chrome)



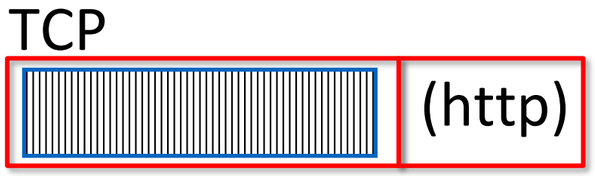
Here it goes….


Ethernet data
IP data
TCP data
(http)
IP Address
(TCP)
Ethernet
“MAC” address
(IP)

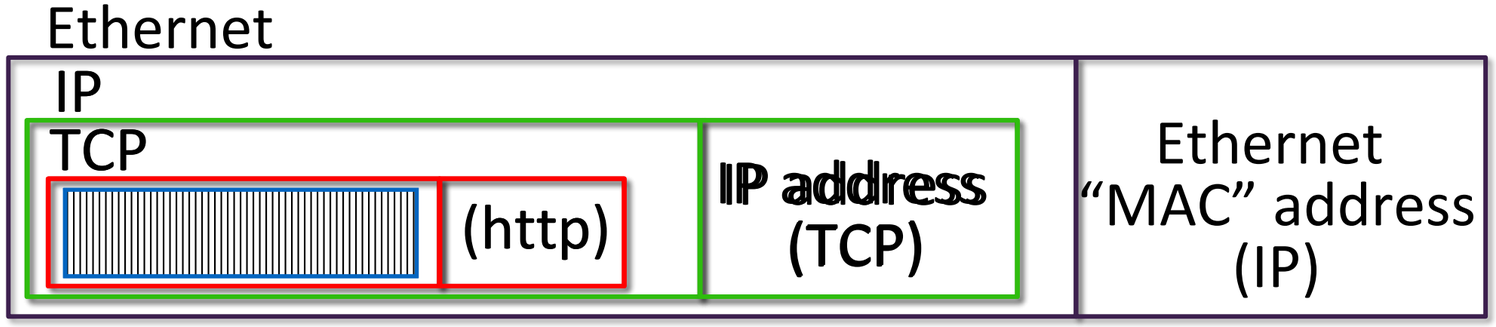
http server (e.g. www.google.com)

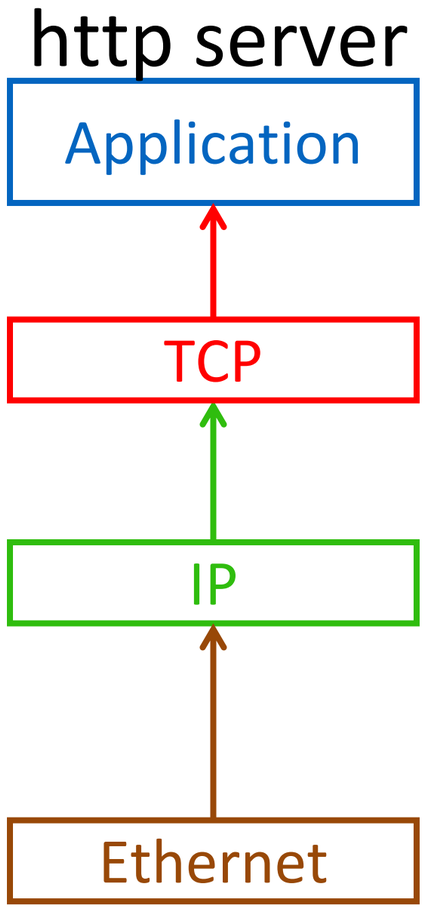
http request

"Deliver to the http server"



"Deliver to TCP"



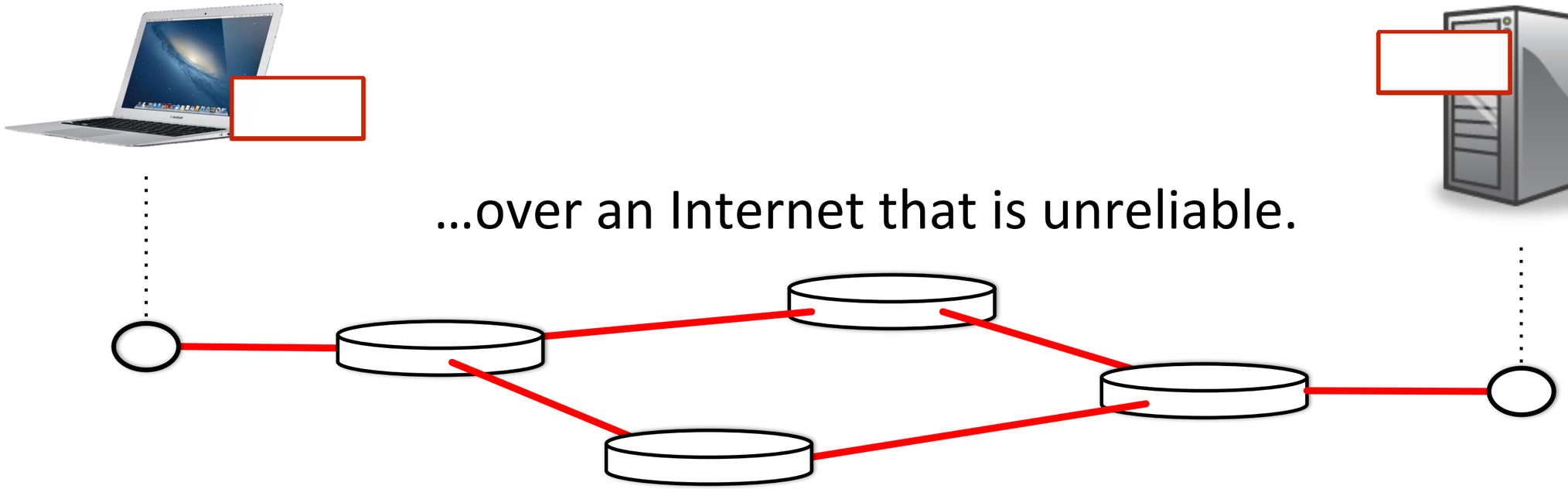


Summary of what we

My Program

Someone else's Program

Applications send and receive data in packets….



…over an Internet that is unreliable.

Packets are forwarded hop-by-hop using the IP destination address.

Our applications use TCP to make sure they are delivered and put back in the correct order.